ハイパフォーマンス
コンピューティング 74－6
（1998．12．11）

# 級数の集約による多倍長数の計算法と $\pi$ の計算への応用

右田剛史 † 天野 晃 † 浅田尚紀 † 藤野清次 ‡

広島市立大学 情報科学部
† 知能情報システム工学科， ‡ 情報工学科

E-mail: {migita, a-amano, asada}@cv.its.hiroshima-cu.ac.jp
E-mail: fujino@ce.hiroshima-cu.ac.jp

**概要**: 多数桁の数学定数，特に $\pi$ や自然対数の計算法として簡単に導出できる級数展開を用いる方法と，$\pi$ における Gauss-Legendre の公式等の反復計算法が知られている．$\pi$ に関しては，従来 $N$ 桁の値を得る計算量は，級数によると $O(N^2)$，反復計算法によると $O(N(\log N)^2)$ とされ，$N$ が大きい時には反復計算法の方が格段に有利であると言われていた．本稿ではある種の級数に対して，隣接する級数の項を集約することにより，$O(N(\log N)^3)$ の計算量で級数の和を計算する計算法を示した．この方法によって桁数 $N$ が大きい時にも，従来計算時間的に反復計算法より不利とされてきた級数による計算が，同等の時間で行える．本手法を用いることにより，3.2 万桁から 5.3 億桁の $\pi$ の計算に関して，級数の和を用いた Chudnovsky の公式を，反復計算による Gauss-Legendre の公式よりも高速に計算できることが明らかになった．

キーワード:
級数の集約, $\pi$の計算, Chudnovsky の公式

# Recursive Reduction of Series for Multiple-precision Evaluation and its Application to Pi Calculation

Tsuyoshi Migita† Akira Amano† Naoki Asada† Seiji Fujino‡

†Dept. of Intelligent Systems, ‡Dept. of Computer Engineering
Hiroshima City University

**Abstract:** Multiple-precision mathematical constants, such as $\pi$ or $e$ are known to be calculated by sum of series. On the other hand, much faster calculation method that use iteration are known for some constants such as $\pi$. For the case of $\pi$, $N$ digits calculation time by method of sum of series is said to be $O(N^2)$, and that of iterational method is $O(N(\log N)^2)$. Thus, for large $N$, iterational method is far more efficient than that of sum of series.

In this paper, we propose a fast algorithm of calculating sum of series in $O(N(\log N)^3)$ time by recursively reducing adjacent terms of series. With this algorithm, calculation time of sum of series become comparable to that of iterational method in case of large $N$. Experimental results on calculating 32,000 to 530 million digits of $\pi$ showed that the Chudnovsky formula which uses sum of series can be calculated faster than the Gauss-Legendre method which uses iterational method.

**Keywords:** reduction of series, $\pi$ calculation, Chudnovsky formula

## 1 はじめに

$\pi$ の値を計算する方法として、級数の和を用いる方法(以下、級数法と呼ぶ)と反復計算による方法(以下、反復法と呼ぶ)が知られている。表1にそれらの代表的な公式を示す。

級数法の場合、$N$ 桁の精度で $\pi$ を求めるには計算すべき項数は $N$ に比例する。従来の計算法では、1項ずつ小数に展開して足し合わせる方法を用いることが多く計算量は $O(N^2)$ であった。これに対して反復法の場合、$N$ 桁の精度で $\pi$ を求める計算量は $O(N(\log N)^2)$ である。このため、級数法よりも Gauss-Legendre の公式や Borwein の4次の収束の公式などの反復法を用いる方がはるかに高速であり、最近の $\pi$ の近似値計算では反復法の使用が主流となっている [1, 2, 4]。

しかし、文献 [1] には、具体的な計算方法は示されていないが級数法の計算量を $O(N(\log N)^3)$ にすることが可能であると示唆されている。そこで本稿では、次に示す形の級数において、隣接する2項を統合して項数を半分にする操作を再帰的に繰り返し、最終的に1項に集約するという手法によって、級数の計算を $O(N(\log N)^3)$ の計算量で行う具体的な計算方法を提案する。

$$S_L = \sum_{k=0}^{L-1} \frac{\prod_{l=0}^{k-1} B_l}{\prod_{l=0}^{k} C_l} A_k \quad (1)$$

$$= \frac{1}{C_0}(A_0 + \frac{B_0}{C_1}(A_1 + \frac{B_1}{C_2}(A_2 + \dots \quad (2)$$

ここで、$A_k, B_l, C_l$ は比較的小さな整数(数十桁以下)であり、$B_l < C_l$ とする。

表1の $\pi$ の級数法は、すべて式(1)を含む形に変形できるため、本手法を用いることによって $\pi$ の級数法による計算を反復法と同程度の時間で計算することが可能である。しかも、本稿で述べる級数の高速計算法は、$\pi$ の計算だけでなく三角関数や次に示す指数関数など多くの初等関数に適用可能であり、その応用範囲は広い。

$$e^x = \sum_{k=0}^{\infty} \frac{x^k}{k!}$$

以下、2章では級数の集約について述べ、3章では本手法を $\pi$ の級数法に適用し反復法と比較した結果を示す。さらに, $e$ の値の計算結果についても述べる。

## 2 級数の集約

従来は式(2)に対する計算を次のように行っていた。多倍長の小数 $1/C_0$ を初項とし、順次 $B_k$ を掛け $C_{k+1}$ で割ることによって次の数列を作る。

$$\{\frac{1}{C_0}, \frac{1}{C_0}\cdot\frac{B_0}{C_1}, \frac{1}{C_0}\cdot\frac{B_0}{C_1}\cdot\frac{B_1}{C_2}, \dots\}$$

この数列を作りながら各項に $A_k$ の重みを掛けて総和を求めると、それが級数の和となる。

表1: $\pi$ の計算公式(上段:級数法, 下段:反復法)

| | |
|---|---|
| Machin | $\pi = 16\arctan\frac{1}{5} - 4\arctan\frac{1}{239}, \qquad \arctan x = \sum_{k=0}^{\infty}(-1)^k \frac{x^{2k+1}}{2k+1}$ |
| Ramanujan | $\frac{1}{\pi} = \frac{2\sqrt{2}}{9801}\sum_{n=0}^{\infty}\frac{(4n)!}{(n!)^4}\frac{[1103+26390n]}{396^{4n}}$ |
| Chudnovsky | $\frac{1}{\pi} = 12\sum_{k=0}^{\infty}\frac{(-1)^k(6k)!}{(3k)!(k!)^3}\frac{13591409+545140134k}{(640320)^{3k+3/2}}$ |
| Gauss-Legendre | 初期値:$A_0 = 1,\ B_0 = \sqrt{1/2},\ C_0 = 1$<br>反復:$A_{N+1} = (A_N + B_N)/2,\ B_{N+1} = \sqrt{A_N B_N},\ C_{N+1} = C_N - 2^N(A_N - B_N)^2$<br>$4A_N B_N / C_N$ が$\pi$に収束する。 |
| Borwein | 初期値:$a_0 = 6 - 4\sqrt{2},\ y_0 = \sqrt{2} - 1,$<br>反復:$y_{k+1} = \frac{1-(1-y_k^4)^{1/4}}{1+(1-y_k^4)^{1/4}},\ a_{k+1} = a_k(1+y_{k+1})^4 - 2^{2k+3}y_{k+1}(1+y_{k+1}+y_{k+1}^2)$<br>$a_n$ が $1/\pi$ に収束する。 |

$$
\begin{aligned}
\pi &= 6\arcsin\frac{1}{2} = 6\sum_{k=0}^{\infty}\frac{(2k)!}{2^{2k}(k!)^2}\cdot\frac{1}{2k+1}\cdot\frac{1}{2^{2k+1}} \\
&= 6\left(\frac{1}{2}+\frac{1}{2\cdot3\cdot2^3}+\frac{1\cdot3}{2\cdot4\cdot5\cdot2^5}+\frac{1\cdot3\cdot5}{2\cdot4\cdot6\cdot7\cdot2^7}+\ldots\right) \\
&= \frac{1}{2}\left(6+\frac{6}{2\cdot3\cdot4}\left(1+\frac{3^2}{4\cdot5\cdot4}\left(1+\frac{5^2}{6\cdot7\cdot4}(\ldots)\right)\right)\right) \\
&\approx (6,6,2):(1,3^2,2\cdot3\cdot4):(1,5^2,4\cdot5\cdot4):(1,7^2,6\cdot7\cdot4):(1,9^2,8\cdot9\cdot4):\quad\ldots\quad:0 \\
&= (150,54,48):(192,1225,13440):(521,9801,126720):(1009,38025,524160):0 \\
&= (2026422,66150,645120):(282976569,372683025,66421555200):0 \\
&= (134616819631533750,24652982103750,42849873690624000):0 \\
&= 134616819631533750/42849873690624000 \quad\approx\quad 3.1415919
\end{aligned}
$$

図 1: $\pi = 6\arcsin(1/2)$ の計算例

小数に展開するのではなく分数のまま計算する方法 [2] もあるが、前から 1 項ずつ計算したのでは、桁数 $N$ に対して計算量は $O(N^2)$ のままである。そこで、隣接する 2 項を統合し項数を半分にする操作を再帰的に繰り返すことによって計算量を削減する。

### 2.1 計算方法

式 (2) の $A_k$, $B_k$, $C_k$ を用いて次の式 (3) の数列 $R_k$ を考え、式 (4) のように表記する。

$$
\begin{aligned}
R_k &= \frac{1}{C_k}(A_k + B_k R_{k+1}) && (3) \\
&\equiv (A_k, B_k, C_k) : R_{k+1} && (4)
\end{aligned}
$$

数列の $L$ 項までの和 $S_L$ は、$R_L = 0$ とした時の $R_0$ の値である。要素 $(A_i, B_i, C_i)$ を $\phi_i$ と略記すると、$S_L$ は次のように表現できる。

$$S_L = \phi_0 : \phi_1 : \phi_2 : \phi_3 : \ldots : \phi_{L-1} : 0 \qquad (5)$$

また $R_k$ と $R_{k+2}$ には、次の関係が成立する。

$$
\begin{aligned}
R_k &= \frac{1}{C_k}\left(A_k + \frac{B_k}{C_{k+1}}(A_{k+1} + B_{k+1}R_{k+2})\right) \\
&= \frac{1}{C_k C_{k+1}}((A_k C_{k+1} + B_k A_{k+1}) + \\
&\qquad B_k B_{k+1} R_{k+2}) \\
&= (A_k C_{k+1} + B_k A_{k+1}, \\
&\qquad B_k B_{k+1}, C_k C_{k+1}) : R_{k+2} \\
&= (\phi_k : \phi_{k+1}) : R_{k+2}
\end{aligned}
$$

すなわち、':' は次のような二項演算子とみなすことができる。

$$(a, b, c) : (A, B, C) = (aC + Ab, bB, cC) \qquad (6)$$

ここで、':' は結合的であり任意の順序で計算することができる。したがって、計算すべき項を前半と後半に再帰的に 2 分割すると効率の良い計算ができる。たとえば、計算すべき項が 8 項の時の計算順序は次のようになる。

$$
\begin{aligned}
S_8 &= (((\phi_0 : \phi_1) : (\phi_2 : \phi_3)) : \\
&\qquad ((\phi_4 : \phi_5) : (\phi_6 : \phi_7))) : 0 \qquad (7)
\end{aligned}
$$

この例からも明らかなように、この表記法では項数 $L$ に対して括弧の深さが $\log L$ となる性質がある。

なお、実際の計算では項数が 2 の累乗とは限らない。そのような場合にも、できるだけ項数が均等となるように分割することにする。

図 1に $\pi = 6\arcsin(1/2)$ の先頭 8 項の計算例を示す。ここでは、級数の集約の基本演算 $(a, b, c)$ : $(A, B, C)$ を図 2のように記し、全体の計算を木の形で表現している。

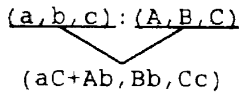


図 2: 集約の基本演算 $(a,b,c):(A,B,C)$

## 2.2 級数の集約の計算量

級数の集約の計算量を評価するために、式 (1) の形式の級数において隣接する 2 項を再帰的にまとめ、1 項に集約することを考える。簡単のため、項数 $L$ は 2 の累乗とする。

式 (1) の 2 項ずつの和を考えると、次のようになる。

$$
\begin{aligned}
S_L &= \sum_{k=0}^{L-1} \frac{\prod_{l=0}^{k-1} B_l}{\prod_{l=0}^{k} C_l} A_k \\
&= \sum_{k=0}^{L/2-1} \frac{\prod_{l=0}^{2k-1} B_l}{\prod_{l=0}^{2k} C_l} \cdot \qquad (8)\\
&\qquad (A_{2k} + \frac{B_{2k}}{C_{2k+1}} A_{2k+1}) \\
&= \sum_{k=0}^{L/2-1} \frac{\prod_{l=0}^{k-1} (B_{2l}B_{2l+1})}{\prod_{l=0}^{k} (C_{2l}C_{2l+1})} \cdot \qquad (9)\\
&\qquad (A_{2k}C_{2k+1} + A_{2k+1}B_{2k}) \\
&= \sum_{k=0}^{L/2-1} \frac{\prod_{l=0}^{k-1} B'_l}{\prod_{l=0}^{k} C'_l} A'_k \qquad (10)
\end{aligned}
$$

式 (10) の $A', B', C'$ は、式 (9) に対応するように置く。式 (10) は式 (8) と同じ形であるので、再び同様の変形を施すことができる。

従って、

$$
\begin{aligned}
A_k^{(0)} &= A_k \\
B_k^{(0)} &= B_k \\
C_k^{(0)} &= C_k \\
A_k^{(n+1)} &= A_{2k}^{(n)} C_{2k+1}^{(n)} + A_{2k+1}^{(n)} B_{2k}^{(n)} \\
B_k^{(n+1)} &= B_{2k}^{(n)} B_{2k+1}^{(n)} \\
C_k^{(n+1)} &= C_{2k}^{(n)} C_{2k+1}^{(n)} \qquad (11)
\end{aligned}
$$

とすると、$A_0^{(n)}/C_0^{(n)}$ は $2^n$ 項までの和となる。

式 (11) は、図 1で行った一連の ':' 演算によって式全体を変形する操作に対応する。この演算には多倍長の乗算と加減算のみが含まれ、1 項に集約された後に除算によって級数の和を計算する。ここで、$N$ 桁の乗算の計算量を $M(N)$ とすると、$O(M(N)) = O(N \log N)$、加減算の計算量は $O(N)$、除算の計算量は $O(M(N))$ である。以下では、加減算や除算の計算量は集約全体の計算量に比べると十分に少ないものとして、多倍長乗算の計算量を評価する。

$A_k, B_k, C_k$ の最大桁数を $K$、計算する項数を $L$ とすると、1 回目の集約は $K$ 桁の乗算を $2L$ 回行うので $CLM(K)$ の計算量となる ($C$ は定数)。2 回目の集約に含まれる乗算は、桁数は倍、回数は半分になるので、$(1/2)CLM(2K)$ の計算量となる。$\log L$ 回の集約によってすべての項は 1 項に集約されるので、この計算量は次のように近似できる。

$$
\begin{aligned}
&CL[\, M(K) + (1/2)M(2K) + \\
&\qquad (1/4)M(4K) + (1/8)M(8K) + \dots\,] \\
= \; &CL[\, (\log L)M(K) + K(\log L)^2/2\,]
\end{aligned}
$$

$N$ 桁の精度で計算する場合、適当な定数 $l, k$ を用いて、計算すべき項数は $L = lN$ となり、$A_k, B_k, C_k$ の最大桁数は $K = k \log N$ となるので、総計算量は $O(N(\log N)^3)$ となる。

## 2.3 級数の ':' 演算子による表現

$\pi$ の級数法による計算公式は、2.1節で導入した ':' 演算子によって表記すると以下のようになる。

Machin の公式

arctan を次のように計算する。

$$
\arctan \frac{1}{p} = (1, -1, p) : (1, -3, 3p^2) : (1, -5, 5p^2) : (1, -7, 7p^2) : \dots : 0
$$

Ramanujan の公式

a=1103, b=26390, c=3073907232 として、

$$
\begin{aligned}
\pi = &\; 9801/(2\sqrt{2})/ \\
&((a, 1 \cdot 1 \cdot 3, 1) : \\
&(a+b, 3 \cdot 5 \cdot 7, c) : \dots : \\
&(a+kb, (2k+1)(4k+1)(4k+3), k^3 c) : \\
&\dots : 0)
\end{aligned}
$$

Chudnovsky の公式

a = 13591409, b = 545140134

c = 10939058860032000 として、

$$
\begin{aligned}
\pi = &\; 426880\sqrt{10005}/ \\
&((a, -1 \cdot 1 \cdot 5, 1) : \\
&(a+b, -3 \cdot 7 \cdot 11, c) : \dots : \\
&(a+kb, -(2k+1)(6k+1)(6k+5), k^3 c) : \\
&\dots : 0)
\end{aligned}
$$

表 2: 級数法による 104 万桁の $\pi$ と $e$ の計算時間 (秒)

| 公式 | 本手法 | 従来法 |
|---|---|---|
| Machin の公式 | 507 | 113443 |
| Ramanujan の公式 | 218 | 73971 |
| Chudnovsky の公式 | 129 | 42631 |
| 自然対数の底 $e$ | 54 | 14320 |

$e$ についても同様に ':' 演算子によって表記すると次のようになる。

$$e = 2 + ((1,1,2):(1,1,3):\ldots:0) \qquad (12)$$

なお、$e$ の値の確認には、次式を用いる。

$$e = ((1,-1,2):(1,-1,3):\ldots:0)^{-1} \qquad (13)$$

## 3 実験と評価

級数の集約によって $\pi$ と $e$ の値を求める実験を行った。実験には次の計算機を用いた。

(i) Gateway G6-200 (PentiumPro 200 MHz, 64MB, FreeBSD 2.2.1R)

(ii) SGI OCTANE (R10000 195MHz, 384MB, IRIX6.4)

なお、級数の集約計算を図 1の様に木の形で表すと、級数の和 $S$ は根にあたる 3 つ組 $(A,B,C)$ の値を使って、$S = A/C$ の除算で与えられる。図 1の例では、$A$ や $C$ の値は 15 桁以上の数値となっているが $S$ は 5 桁の精度しかないように、計算途中で $A$, $B$,$C$ の各値は必要な精度 $N$ 桁よりも大きくなる場合がある。しかし、計算途中で $N$ 桁を越える数値を全て $N$ 桁で切捨てても、根の $A$ と $C$ もほぼ $N$ 桁の精度で求まり、$S$ を $N-\alpha$ 桁まで正しく求めることができる。この $\alpha$ は乗算や除算の精度によって決まる数で、$N$ が十分に大きい時には $N \gg \alpha$ である。

### 3.1 級数の集約の評価

級数の集約による高速化の効果を調べるために、従来の級数計算法 (計算量 $O(N^2)$) と本手法を比較した。$\pi$ の計算は、小数点以下 $2^{20}-4 = 1,048,572$ 桁の精度で行い、計算機 (i) を使用した。表 2および図 3に計算時間を示す。この結果、級数の集約を用いた場合、従来の級数計算法の 1/340 ~ 1/200 の時間で計算できることがわかった。なお、計算誤差は下 48 桁以下であった。

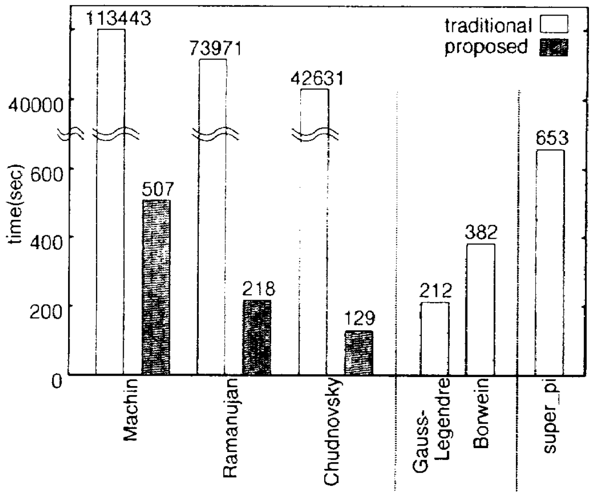


図 3: 104 万桁の $\pi$ の計算時間

同じ桁数の $e$ の計算では、表 2の $\pi$ の計算で最も高速な Chudnovsky の公式の 129 秒と比べてさらに半分以下の時間で計算できることがわかった。これは, $e$ の計算において ':' 演算子表記の 3 つ組の第 2 項が常に ±1 となり、乗算の必要がないためである。なお、式 (12) と式 (13) の結果を照合したところ計算誤差は下 12 桁であった。

### 3.2 級数法と反復法の比較

#### 3.2.1 104 万桁の $\pi$ の計算

$\pi$ の計算の反復法として Gauss-Legendre の公式と Borwein の 4 次の収束の公式を使用した。これらの方法によって約 104 万桁の $\pi$ を計算する実験を計算機 (i) で行った。図 3の右に計算時間を示す。級数の集約を行った級数法の結果と比較すると、Chudnovsky の公式は Gauss-Legendre の公式と Borwein の 4 次の収束の公式のいずれよりも高速に計算し得ることがわかる[1]。なお、反復法による計算誤差は下 120 桁以下であった。

参考のため、高速な $\pi$ 計算プログラムとして知られる super_pi[3] を計算機 (i) で実行し計算時間を計測した。OS や計算桁数の違いがあり単純な比較はできないが、653 秒で 1,048,576 桁まで正しく計算された。

[1] ここで用いた Gauss-Legendre の公式は、平方根計算法等の改善により Borwein の 4 次の収束の公式よりも高速となった。文献 [4] では、この 2 つの公式による計算量はほぼ等しいとされている。

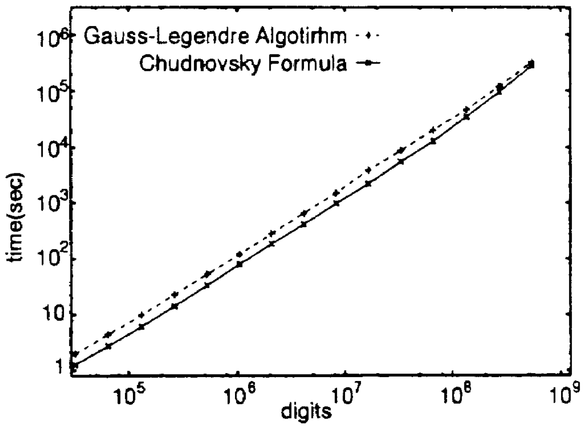


図 4: Chudnovsky の公式と Gauss-Legendre の公式の比較

### 3.2.2 5.3 億桁の $\pi$ の計算

級数の集約による Chudnovsky の公式の計算量は $O(N(\log N)^3)$ であり、Gauss-Legendre の公式の計算量 $O(N(\log N)^2)$ に比べて $\log N$ が掛かっているため、計算桁数の増大によって不利になることが予想される。

そこで両公式による 3.2 万桁から 5.3 億桁までの $\pi$ の計算時間を比較した。なお、計算機 (i) では資源が不足するため計算機 (ii) を用いた。図 4 に桁数と計算時間の関係を、表 3 に 104 万桁と 5.3 億桁の計算時間を示す。5.3 億桁の計算では、両公式で $2^{29}-4=536,870,908$ 桁目まで計算した結果、誤差は下 236 桁であった。

この実験の結果、1 億桁以下の計算では Gauss-Legendre の公式は Chudnovsky の公式の平均 1.56 倍の時間を要し、桁数による時間比率の変化はほとんど見られなかった。しかし 1 億桁以上では両公式による計算時間の差が少なくなった。これは、主に乗算桁数の違いによるものである。$N$ 桁の $\pi$ を計算する際には、今回用いた Gauss-Legendre の公式では $N/8$ 桁の乗算を行えば十分であるのに対して、級数の集約では $N$ 桁の乗算を行う必要がある。しかし計算機 (ii) では、主記憶容量の制限により 1 億桁を越える乗算を FFT によって高速に行うことはできないため、遅い方法を使わざるを得なかった。1 億桁を越える桁数での速度低下を抑える方法は現在検討中である。

なお、計算機 (ii) 上で、式 (12) を使い $e$ の値を 536,870,788 桁計算するのに 12 時間 38 分を要した。

表 3: 計算機 (ii) による $\pi$ と $e$ の計算時間

| 公式 | 104 万桁 (分) | 5.3 億桁 (時間) |
|---|---|---|
| Chudnovsky の公式 | 1.33 | 75.05 |
| Gauss-Legendre の公式 | 2.01 | 88.77 |
| 自然対数の底 $e$ | 0.56 | 12.63 |

## 4 まとめ

数学定数や初等関数の計算法として、級数を用いる方法と一部ではあるが効率の良い反復計算による方法とが知られている。一般に級数の計算は反復計算法に比べ計算時間的に不利とされていた。本稿では、この多倍長級数の高速な計算法を提案し、従来法では $O(N^2)$ の計算量であった $N$ 桁の計算を $O(N(\log N)^3)$ の計算量で行えることを示した。

本手法の有効性を示すために例として $\pi$ と $e$ の計算を取り上げ実験を行った。従来 $\pi$ の値を計算する場合、級数よりも Gauss-Legendre の公式等の反復計算法を用いる方が格段に有利とされていたが、級数の集約を行うことによって同程度の計算時間となることがわかった。

## 参考文献


[1] Bailey, J.Borwein, P.Borwein: Ramanujan, Modular Equations, and Approximations to Pi or How to compute One Billion Digits of Pi, http://www.cecm.sfu.ca/organics/papers/borwein.

[2] 金田: $\pi$ のはなし, 東京図書, 1991.

[3] ftp://www.cc.u-tokyo.ac.jp

[4] 高橋、金田: 分散メモリ計算機による円周率の 515 億桁計算, 情処学論 Vol.39, pp. 2074–2083 (1998).